하계학술회의
논문85-8-5

2의 보수를 이용한 병렬승산기의 개선

Improvement of Parallel Multiplier using 2's complement

김 원 섭
조 해 룡
× 김 신 배

진 북 대 학 교
전 기 공 학 과

1. 서 론

컴퓨터에서 승산(Multiplication)은 Digital System 의 가장 기본적인 연산이라 할 수 있다.

초기의 순차적인 Add-and-Shift 승산기보다 좀더 빠른 속도를 얻기 위하여 병렬승산기를 사용하게 되었다. 병렬승산기(Parallel Multiplier)를 사용함에 있어서 음수(-)를 곱할때 2의 보수표현이 바람직한데 2의 보수표현에서는 부호 bit가 그 숫자 자체에 포함되어 있기 때문에 부호 bit 처리에 많은 어려움을 겪었다. 1), 2)

1971년 Pezaris는 4가지 Type (0 Type, 1 Type, 2 Type, 3 Type)의 cell을 이용함으로서 2의 보수표현에 의한 편리한 승산기를 설계하였다. 3) 1973년 Baugh와 Wooley는 승산 Matrix를 알맞게 변화시켜 4가지 Type 중에서 오직 한가지 Type(0 Type)으로만 2의 보수표현에 의한 승산기의 알고리즘을 개발하여 승산기의 구성을 단순화 하였다. 4) 그러나 Baugh-Wooley 승산기는 1 가지 Type으로 단순화 하였지만 cell의 수가 약간 증가했으며 사전에 승수와 피승수의 보수처리과정이 필요하기 때문에 좀 복잡했다. 1983년 Bandeira, Vaccaro, Haward 는 Baugh-Wooley승산기보다 더 간단하고 보수처리가 필요치않는 2가지 Type(1 Type, 2 Type)만으로 된 승산기를 설계하였다. 5)

본 논문에서는 0 Type, 2 Type 만을 사용하여 Baugh-Wooley 승산기에서 필요했던 보수처리가 필요치않고 더 간략화되어 개선된 승산기를 설계하였다.

2. 본 론

1) 2의 보수를 이용한 승산기의 기본 원리

승산에 있어서 부호의 처리는, 기본적인 승산법에서는 부호를 생략하거나 수의 승산이끝난 후에 부호 bit를 exclusive-OR 게이트로서 처리하여 부호를 결정하는 방법을 사용해왔다.

그러나 2의 보수를 이용하면 따로 부호 bit를 처리할 필요없이 부호 bit와 숫자를 같이 승산기로 처리하여 원하는 결과를 얻을수 있다. 예를 들면, $A=(a_{n-1}\,a_{n-2}\cdots\cdots a_1\,a_0)$ 를 2의 보수화한 수라 하자. 여기서 a 는 부호 bit이다.

실제의 이 값을 10진수로 표현하면

$$A=-a_{n-1}\cdot 2^{n-1}+\sum_{i=0}^{n-2}a_i\cdot 2^i \quad \text{----------(1)}$$

라 쓸수 있다.

식 (1)은 (-)의 값을 갖는 $(-a_{n-1}\cdot 2^{n-1})$ 와 (+)값을 갖는 $\sum_{i=0}^{n-2}a_i\cdot 2^i$ 로 나뉘어 진다. (+)값과 (-)값을 구분하기 위하여 (-)값에는 괄호를 의워 A를 $(a_{n-1})\,a_{n-2}\cdots\cdots a_1\,a_0$ 로 표현하기로 하여 임의의 수 A, B에 대한 승산 Matrix를 그림 1과 같이 나타낼 수 있다.

$$\begin{array}{cccccccc}
 & & & & (a_3) & a_2 & a_1 & a_0 \\
 & & & & (b_3) & b_2 & b_1 & b_0 \\
\hline
 & & & & (a_3b_0) & a_2b_0 & a_1b_0 & a_0b_0 \\
 & & & (a_3b_1) & a_2b_1 & a_1b_1 & a_0b_1 & \\
 & & (a_3b_2) & a_2b_2 & a_1b_2 & a_0b_2 & & \\
 & a_3b_3 & (a_2b_3) & (a_1b_3) & (a_0b_3) & & & \\
\hline
P_7 & P_6 & P_5 & P_4 & P_3 & P_2 & P_1 & P_0
\end{array}$$

그림 1. 기본적인 승산 Matrix. (4×4)

2) 보수처리를 위한 4가지 Type의 cell

그림 1의 Matrix를 계산하기 위해서 기본적인 Full Adder를 사용 할 수 없다. 왜냐하면 Full Adder는 모든 입력이 (+)값을 갖아야 하지만 그림 1의 Matrix에는 (-)값이 존재하기때문이다. 그러므로 (-)값도 입력, 출력이 될 수 있는 다음 4가지 Type의 cell을 사용해야 한다.

각각 Type의 출력값은

0 Type: $S=\bar{x}\bar{y}z+\bar{x}y\bar{z}+x\bar{y}\bar{z}+xyz$
$C=xy+yz+zx$

1 Type: $S=\bar{x}\bar{y}z+\bar{x}y\bar{z}+x\bar{y}\bar{z}+xyz$
$C=\bar{x}y+yz+\bar{x}z$

2 Type: $S = \bar{x}\bar{y}z + \bar{x}y\bar{z} + x\bar{y}\bar{z} + xyz$

$C = xy + y\bar{z} + x\bar{z}$

3 Type: $S = \bar{x}\bar{y}z + \bar{x}y\bar{z} + x\bar{y}\bar{z} + xyz$

$C = xy + yz + xz$

Pezaris 는 그림 1의 Matrix를 이 4가지 Type으로 직접 구성했고, Baugh와 Wooley는 그림 1의 Matrix를 0 Type으로만 구성할 수 있도록 개선하였으며 Bandeira, Vaccaro, Haward는 Matrix를 바꿔 1 Type, 2 Type으로서 구성하였다.

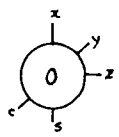
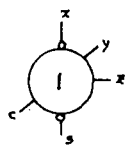
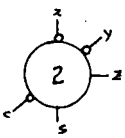
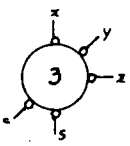
0 Type cell		x y +) z c s
1 Type cell		- x y +) z c (-s)
2 Type cell		- x - y +) z (-c) s
3 Type cell		- x - y +) - z (-c)(-s)

그림 2. 4가지 Type의 cell.

3) 기존 승산기의 구조

Baugh-Wooley승산기는 0 Type으로서만 구성되었지만 보수처리가 되어야 하고 4x4 bit의 경우 Pezaris승산기보다 3개의 cell이 더 필요한 것을 볼수 있다.(그림 3),(그림 4).또한, Bandeira, Vaccaro, Haward는 그림 1의 기본 Matrix를 알맞게 변화시켜 2가지 Type만을 사용한 승산기를 구성하였다.(그림 5) 그 결과, 4가지 Type을 사용한 Pezaris승산기에비해 2가지 Type을 사용함으로서 간략화 되었고, 1가지 Type을 사용한 Baugh-Wooley 승산기에 비해서는 2가지 Type을 사용했지만 2의 보수가 필요치 않아졌고 cell의 수도 감소하였다. 그러나 이 승산기도 그림 1의 Matrix를 그에 맞게 고쳐서 승산을 해야했다.

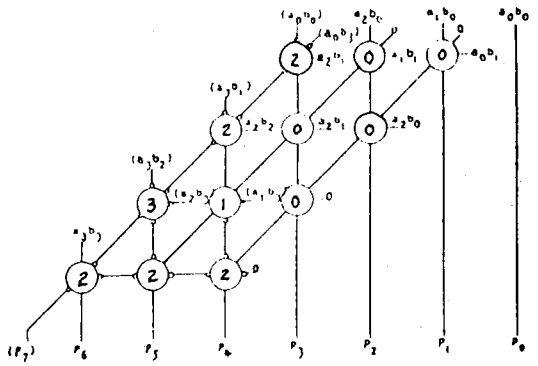


그림 3. Pezaris 승산기. (4x4) bit.

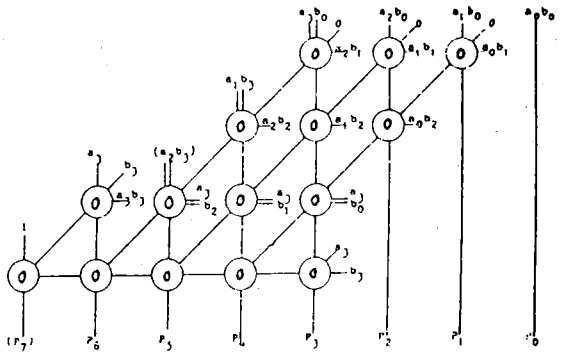


그림 4. Baugh-Wooley 승산기.

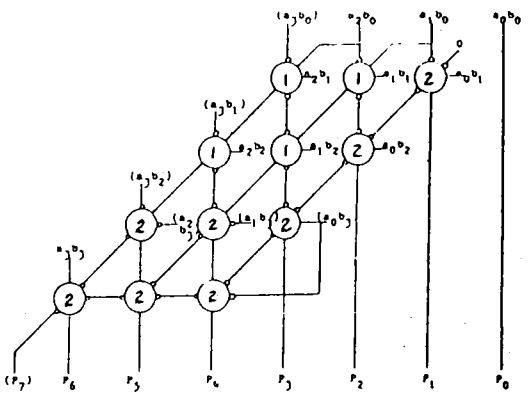


그림 5. Bandeira-Vaccaro-Haward 승산기.

4) 개선된 승산기

본 논문에서는 0 Type과 2 Type을 사용하여 그림 1의 기본승산 Matrix를 변화시키지 않고 직접 적용할 수 있는 승산회로를 구성하였다.

그림 6에서 보는바와 같이 m×n bit의 경우 전체 cell의 수가 m(n-1)개의 cell이 필요한 Pezaris승산기와 Bandeira,Vaccaro,Haward승산기와 같고 m(n-1)+3 개의 cell이 필요한 Baugh,Wooley승산기보다 작으며,보수처리가 필요치 않으면서 cell의 배열이 단순해졌음을 알 수 있다.

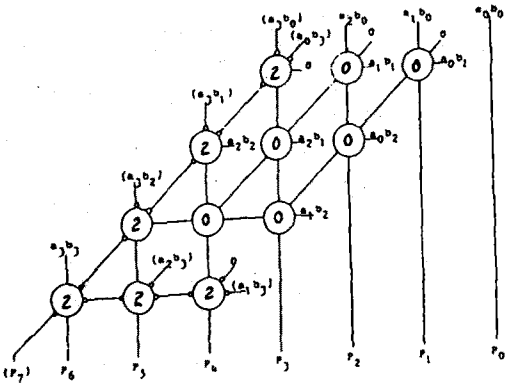


그림 6. 개선된 승산기.

3.결 론

2의 보수를 이용한 승산기들은 계속하여 발전을 거듭하여 왔다.

특히,부호 bit 처리에 있어서 여러가지 난제를 지닌 2의 보수를 이용한 승산기는 Pezaris가 4가지 Type의 cell을 이용한 알고리즘을 설계하였고, 그 후 Baugh와 Wooley가 승산 Matrix를 (+)의 수로 모두 바꾼 0 Type을 이용한 승산기를 설계하였고 Bandeira,Vaccaro,Haward 는 1 Type cell과 2Type cell을 이용한 승산기를 설계하였다.

그러나 이들 승산기들은 모두 장점이 있는 대신에 한두가지씩의 단점을 지니고 있어 본 논문에서는 0Type cell 과 2Type cell을 이용하여 장점은 그대로 간직한채 단점을 좀더 보완한 새로운 승산기를 설계하였다. 앞으로 한가지 cell을 사용하면서 승산 Matrix를 그대로 적용 할 수 있는 승산기의 개발이 요구된다.

참 고 문 헌

(1) M.M.Mano.Computer System Architecture.pp.366-372.Prentice-Hall,Inc.

(2) A.Habbi and P.A.Wintz."Fast multipliers".IEEE Trans.Comput.,vol.C-19.pp.153-157.Feb.1969.

(3) S.D.Pezaris."40ns 17-bit by 17-bit array multiplier".IEEE Trans.Comput.,vol.C-20.pp.442-447.Apr.1971.

(4) C.R.Baugh and Wooley."A two's complement parallel array multiplication algorithm".IEEE Trans.Comput.,vol.C-22.PP.1045-1047.Dec.1973.

(5) N.Bandeira,K.Vaccaro and J.A.Haward."A Two's Complement Array Multiplier Using True Values of the Operands".IEEE Trans.Comput.,vol.C-32.pp.745-747.Aug.1983.